PROGRAMACIÓN DE SISTEMAS DE TIEMPO REAL

PRACTICA 1.
Programación de dispositivos y planificador cíclico simple

Tutor: Javier Fernández Muñoz

(jfernand@arcos.inf.uc3m.es)

Apartado A)

Se desea construir el sistema electrónico de una carretilla controlada por computador desde un servidor externo. La misión de dicha carretilla es transportar cemento al interior de una mina. Para desplazarse dispone de un raíl dispuesto a lo largo del terreno por el que debe avanzar.

1 Características del entorno a controlar:

Por una parte el raíl permite despreciar el rozamiento por lo que en terreno llano mantiene la velocidad. En cambio cuando debe afrontar terreno empinado la carretilla sufrirá una aceleración o deceleración según deba afrontar una subida o bajada. La siguiente tabla indica los cambios en la velocidad debido al terreno.

TERRENO	ACELERACIÓN
Llano	0 m/s^2
Subida	-0,25 m/s^2
Bajada	0,25 m/s^2

El sistema dispone de un sistema GPS que permite conocer con antelación si el terreno en los próximos metros será llano, o tendrá una pendiente de subida o bajada.

Para mantener el control de la caretilla mientras avanza por el rail lo más rápido posible es necesario controlar que la velocidad se mantenga próxima a 55 m/s. Se considera como segura cualquier velocidad entre 40m/s y 70 m/s.

Para controlar la velocidad la carretilla dispone de un mecanismo acelerador que permite incrementar la velocidad en 0,5 m/s^2. Además dispone de un mecanismo de frenado que permite reducir la velocidad en 0,5 m/s^2.

Así mismo la carretilla incluye un deposito de cemento con un mezclador que evita que el cemento fragüe. El cemento debe removerse con el mezclador durante al menos 30 segundos consecutivo, y luego se puede dejar reposar como máximo 60 segundos antes de tener que volver a activar el mezclador.

Por otro lado el motor del mezclador no permite que este activado más de 60 segundos consecutivos, y una vez activado debe reposar por al menos 30 segundos antes de poder volver a funcionar.

2 Definición del sistema a controlar:

El sistema de control consta de dos partes diferenciadas:

- El sistema electrónico de control que esta instalado en la propia carretilla
- El sistema de control remoto instalado en un servidor autónomo del centro de control.

Ambos sistemas se comunican utilizando una conexión serie UART estándar.

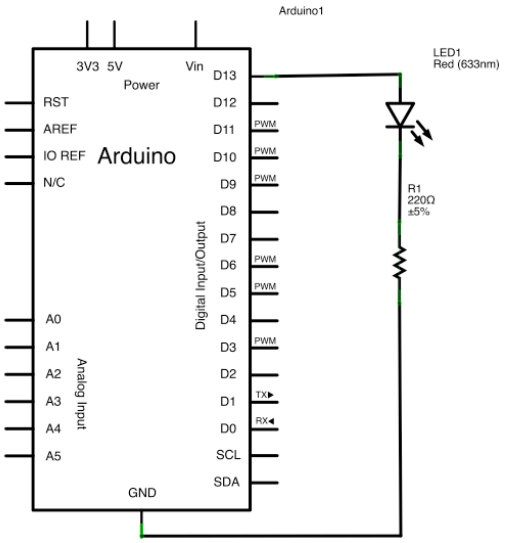
2.1 Definición del sistema electrónico de control de la carretilla:

El sistema electrónico de la caretilla cuenta con diversos sensores y activadores para recibir la información del entorno e interactuar con el.

El hardware del sistema electrónico constará de un placa de microcontrolador modelo Arduino UNO. Los elementos sensores y activadores se describen a continuación.

2.1.1 Activación/desactivación del sistema de aceleración:

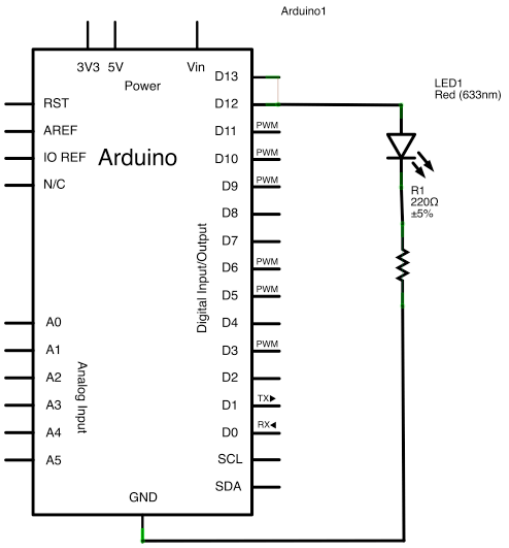
El sistema de aceleración debe implementarse mediante un LED que se encenderá cuando se active el acelerador y se apagara en caso contrario. El esquema del circuito necesario es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que compruebe cada periodo el valor actual que debe tener el acelerador y que active o desactive el LED según corresponda. El alumno deberá elegir un valor apropiado para dicho periodo.

2.1.2 Activación/desactivación del sistema de frenado:

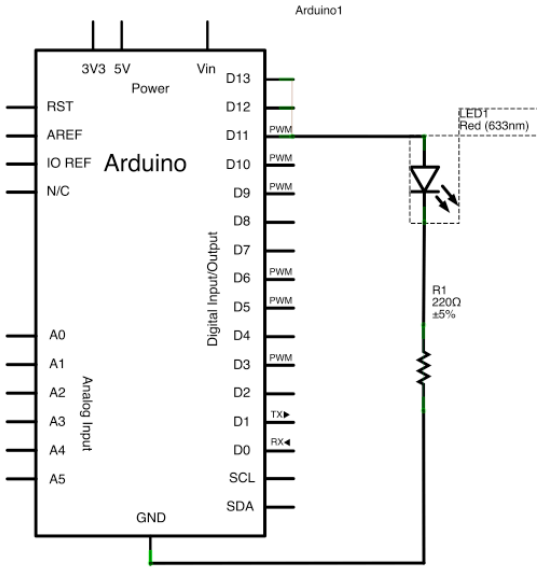
El sistema de frenado debe implementarse mediante un LED que se encenderá cuando se active el freno y se apagara en caso contrario. El esquema del circuito necesario es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que compruebe cada periodo el valor actual que debe tener el freno y que active o desactive el LED según corresponda. El alumno deberá elegir un valor apropiado para dicho periodo.

2.1.3 Activación/desactivación del sistema de mezclado:

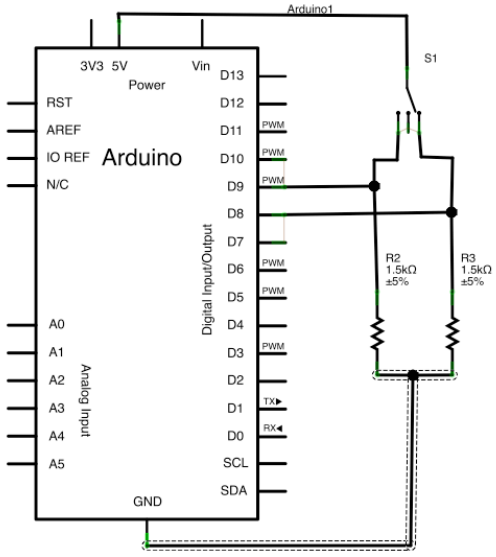
El sistema de mezclado debe implementarse mediante un LED que se encenderá cuando se active el mezclador y se apagara en caso contrario. El esquema del circuito necesario es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que compruebe cada periodo el valor actual que debe tener el mezclador y que active o desactive el LED según corresponda. El alumno deberá elegir un valor apropiado para dicho periodo.

2.1.4 Lectura de la pendiente del rail

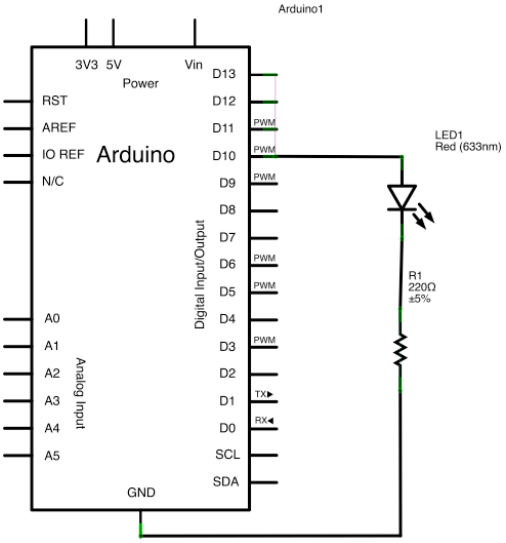
El sistema de recogida del valor pendiente será implementado con un switch de tres posiciones (o tres switches DIP) que el alumno podrá activar.



Deberá programarse una tarea periódica en el microcontrolador que compruebe cada periodo el valor actual de cada posición del switch y así determinar cual es la pendiente actual (solo será valido cuando haya un solo valor). Solo se podrá reseñar si hay subida y si hay bajada. El tramo llano se representa con la ausencia de subidas y bajadas. El alumno deberá elegir un valor apropiado para dicho periodo.

2.1.5 Representación de la velocidad actual:

La velocidad debe representarse con un LED que varíe su nivel de brillo según la velocidad actual. EL LED deberá apagarse cuando llegue a una velocidad de 40 m/s y alcanzará el máximo brillo cuando llegue a 70 m/s. El esquema del circuito necesario es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que calcule periódicamente el valor correcto de la velocidad y que lo represente variando el brillo del LED como corresponda. El alumno deberá elegir un valor apropiado para dicho periodo.

2.1.6 Implementación del servidor de comunicaciones:

El microcontrolador Arduino deberá incluir una tarea que, periódicamente, cada 200 milisegundos compruebe si se ha recibido una petición desde el sistema remoto. En caso afirmativo deberá realizar la acción correcta y enviar la respuesta correcta.

Tanto los mensajes de envío como los de recepción contendrán 9 caracteres (incluido el fin de línea). Esto permite detectar cuando se ha recibido un mensaje completo antes de empezar a leerlo.

Hay dos posibles tipos de comandos que se pueden recibir del sistema remoto.

a) Petición de lectura del valor de un sensor.
   En este caso el sistema remoto envía un mensaje de petición indicando el sensor al que quiere acceder. El servidor de comunicaciones le envía un mensaje con el valor actual de dicho sensor que hay almacenado en la memoria del Arduino.
b) Petición de activar un elemento hardware.
   En este caso el sistema remoto envía un mensaje indicado el elemento HW y si se quiere activar (SET) o desactivar (CLR). El servidor guardara la orden en memoria para que la tarea correspondiente la lleve a cabo y mandara un mensaje de OK.

En caso de que el mensaje enviado del sistema remoto no se pueda entender o haya cualquier tipo de error el servidor de comunicaciones enviará un mensaje de error.

La siguiente tabla incluye Todos los mensajes del protocolo para cada tarea.

Petición	Mensaje petición	Mensaje respuesta	Mensaje error
Activar Acelerador	GAS:<SP>SET<CR> GAS:<SP>CLR<CR>	GAS:<SP><SP>OK<CR>	MSG:<SP>ERR<CR>
Activar Freno	BRK:<SP>SET<CR> BRK:<SP>CLR<CR>	BRK:<SP><SP>OK<CR>	MSG:<SP>ERR<CR>
Activar Mezclador	MIX:<SP>SET<CR> MIX:<SP>CLR<CR>	MIX:<SP><SP>OK<CR>	MSG:<SP>ERR<CR>
Leer Pendiente	SLP:<SP>REQ<CR>	SLP:<SP><SP>UP<CR> SLP:DOWN<CR> SLP:FLAT<CR>	MSG:<SP>ERR<CR>
Leer Velocidad	SPD:<SP>REQ<CR>	SPD:00.0<CR>	MSG:<SP>ERR<CR>

2.2 Definición del sistema remoto:

El sistema remoto se encarga de obtener los datos del sistema electrónico de la carretilla, toma las decisiones correspondiente y envía las peticiones de actuación de vuelta a la carretilla.
El sistema remoto y la carretilla estarán conectados por una conexión seria UART

Además de lo anterior el sistema remoto dispondrá de un display para mostrar por pantalla el estado actual del sistema. Dicho display será accesible mediante un conjunto de llamadas a funciones.

Las tareas que debe realizar el sistema son las siguientes:

2.2.1 Lectura de la pendiente actual:

El sistema remoto deberá implementar una tarea periódica que envíe una petición a la carretilla para recibir el valor actual de la pendiente. Una vez recibida, la tarea mostrar la pendiente por el display utilizando la función correspondiente.
El periodo de esta tarea debe ser de 10 segundos.

2.2.2 Lectura de la velocidad actual:

El sistema remoto deberá implementar una tarea periódica que envíe una petición a la carretilla para recibir el valor actual de la velocidad. Una vez recibida, la tarea mostrar la velocidad por el display utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

2.2.3 Activar el acelerador:

El sistema remoto deberá implementar una tarea periódica que calcule a partir a partir de los datos recibidos si debe activar el acelerador o no. Una vez hecho deberá enviar una petición a la carretilla para activar o desactivar el acelerador. Una vez hecho, la tarea mostrará por el display si el acelerador esta activo o no utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

2.2.4 Activar el freno:

El sistema remoto deberá implementar una tarea periódica que calcule a partir a partir de los datos recibidos si debe activar el freno o no. Una vez hecho deberá enviar una petición a la carretilla para activar o desactivar el freno. Una vez hecho, la tarea mostrará por el display si el freno esta activo o no utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

2.2.5 Activar el Mezclador:

El sistema remoto deberá implementar una tarea periódica que calcule a partir a partir de los datos recibidos si debe activar el mezclador o no. Una vez hecho deberá enviar una petición a la carretilla para activar o desactivar el mezclador. Una vez hecho, la tarea mostrará por el display si el mezclador esta activo o no utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

Las funciones de acceso al display del servidor remoto son las siguientes:

ELEMENTO	*FUNCIÓN*	*TIEMPO DE COMPUTO*
Pendiente	displaySlope(int slope)	0.5 Segundos
Velocimetro	displaySpeed(double speed)	0.5 Segundos
Acelerador	displayGas (int gas)	0.5 Segundos
Freno	displayBrake (int brake)	0.5 Segundos
Mezclador	displayMix (int mixer)	0.5 Segundos

Se pide:

1. Diseñar un planificador cíclico que permita controlar el sistema descrito anteriormente Para:
    a) El sistema remoto
    b) El control electrónico de la carretilla

2. Implementar dicho planificador cíclico para:
    a) El sistema remoto utilizando el SSOO VxWorks y el código de apoyo
    b) El control electrónico de la carretilla utilizando el microcontrolador Arduino UNO y hardware de apoyo.

El código de apoyo para el sistema a controlar se encuentra en **practica_1_sketch_2020-v1.zip**. Este contiene los siguientes ficheros:

- sketch.pde: Fichero de ejemplo del sketch para Arduino del sistema a controlar.

El código de apoyo para el sistema remoto se encuentra en **practica_1A_vxworks_2020-v1.zip**. Este contiene los siguientes ficheros:

- displayA.c: Incluye el código principal del display del apartado A.
- displayA.h: Incluye la cabecera de las funciones de control de display del apartado A.
- controladorA.c: Fichero de ejemplo para realizar el controlador del apartado A. Este fichero debe ser modificado y entregado como parte de la práctica.

Además se incluirá el código de apoyo de un modulo para conectar el sistema a controlar y el sistema remoto mediante un cable serie (**modulo_serie-v1.zip**). Este contiene los siguientes ficheros:

- Serialdll.dll: librería dinámica para Windows que permite comunicar el Arduino y el PC mediante un puerto serie COM.
- serialcall.c: Código de un modulo kernel que permite exportar la funcionalidad de la librería anterior al kernel de VxWorks.
- serialcallLib.c: Codigo que permite enlazar una aplicación VxWorks con el modulo de kernel anterior (a incluir en el programa del sistema remoto).
- serialcallLib.h: Fichero de cabeceras del código de enlace anterior (a incluir en el programa del sistema remoto).

Apartado B)

Se desea ampliar el sistema del apartado A para incluir el manejo automático de los focos de la carretilla.

1 Características del entorno a controlar:

El sistema a controlar es el mismo del apartado A pero con la inclusión de zonas de oscuridad (cuevas) donde será necesario activar los focos de la carretilla para poder tener visibilidad. Estos focos deben activarse automáticamente cuando se detecte oscuridad y desactivarse cuando la carretilla este a la luz del día.

El objetivo es activar los focos tan pronto se entre en una zona de penumbra. Al mismo tiempo deben desactivarse tan pronto se salga de dicha zona. Si no hay cambios no es necesario modificar su estado. El sistema debe responder a los cambios tan pronto como sea posible, pero como mínimo debe responder antes de 12 segundos.

Los nuevos elementos a controlar son:

- Sensor de luminosidad: Determina si el entorno esta suficientemente oscuro para encender las luces o no.
- Focos: Activados permiten la visibilidad en zonas de cuevas. Deben desactivarse al aire libre para ahorrar.

2 Definición del sistema a controlar:

Al igual que en el apartado A sistema de control consta de dos partes diferenciadas:

- El sistema electrónico de control que esta instalado en la propia carretilla
- El sistema de control remoto instalado en un servidor autónomo del centro de control.

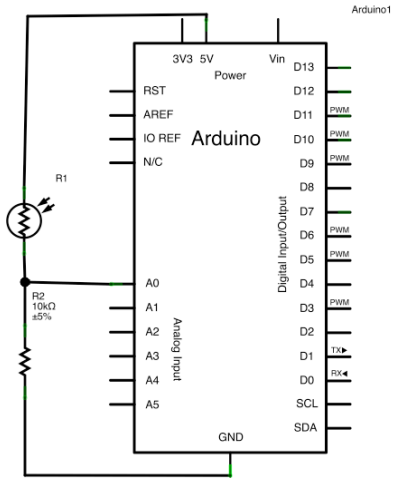
Ambos sistemas se comunican utilizando una conexión serie UART estándar.

2.1 Definición del sistema electrónico de control de la carretilla:

Los requisitos del sistema electrónico de la caretilla será los del apartado A pero añadiendo un sensor de luminosidad y un activador de los focos. Los elementos sensores y activadores añadidos se describen a continuación.

2.1.1 Lectura del sensor de luminosidad:

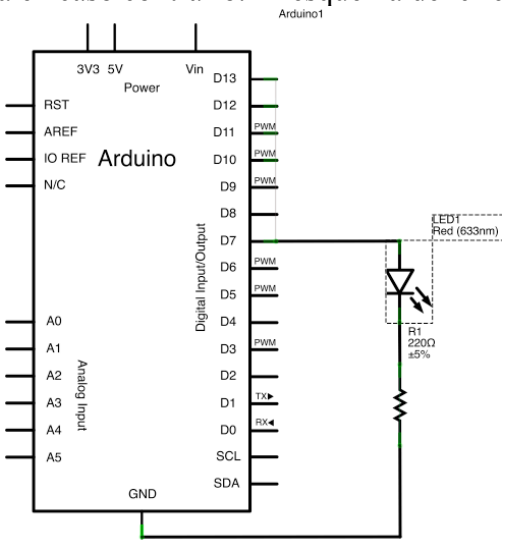
El sensor de luminosidad se implementa mediante una resistencia fotosensible (LDR). El esquema del circuito necesario es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que compruebe cada periodo el valor actual que tiene el sensor. El alumno deberá elegir un valor apropiado para el periodo.

2.1.2 Activación/desactivación de los focos:

El sistema de aceleración debe implementarse mediante un LED que se encenderá cuando se active los focos y se apagara en caso contrario. El esquema del circuito es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que compruebe cada periodo el valor actual que debe tener el acelerador y que active o desactive el LED según corresponda. El alumno deberá elegir un valor apropiado para dicho periodo.

2.1.3 Implementación del servidor de comunicaciones:

El servidor de comunicaciones deberá modificarse para aceptar dos nuevas peticiones
La siguiente tabla incluye todos los mensajes del protocolo para las nuevas tareas.

Petición	Mensaje petición	Mensaje respuesta	Mensaje error
Leer sensor luz	LIT:<SP>REQ<CR>	LIT:<SP>00%<CR>	MSG:<SP>ERR<CR>
Activar Focos	LAM:<SP>SET<CR> LAM:<SP>CLR<CR>	LAM:<SP><SP>OK<CR>	MSG:<SP>ERR<CR>

2.2 Definición del sistema remoto:

El sistema remoto es similar al del apartado A pero con dos nuevas tareas:

2.2.1 Lectura del sensor de luz:

El sistema remoto deberá implementar una tarea periódica que envíe una petición a la carretilla para recibir el valor actual del sensor de luz. Este valor se recibira como un porcentaje entre 0% y 99% y mostrará el valor por el display utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

2.2.2 Activar /desactivar los focos:

El sistema remoto deberá implementar una tarea periódica que calcule a partir a partir de los datos recibidos si debe activar los focos o no (el umbral de luminosidad debe ser el 50%). Una vez hecho deberá enviar una petición a la carretilla para activar o desactivar los focos. Al final, la tarea mostrará por el display si los focos esta activo o no utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

Las funciones de acceso al display del servidor remoto son las siguientes:

ELEMENTO	***FUNCIÓN***	***TIEMPO DE COMPUTO***
Sensor lumniosidad	displayLightSensor (int isDark)	0.5 Segundos
Focos	displayLamps (int light)	0.5 Segundos

Se pide:

3. Diseñar un planificador cíclico que permita controlar el sistema descrito anteriormente Para:
    c) El sistema remoto
    d) El control electrónico de la carretilla

4. Implementar dicho planificador cíclico para:
    c) El sistema remoto utilizando el SSOO VxWorks y el código de apoyo
    d) El control electrónico de la carretilla utilizando el microcontrolador Arduino UNO y hardware de apoyo.

El código de apoyo para el sistema a controlar se encuentra en **practica_1_sketch_2020-v1.zip**. Este contiene los siguientes ficheros:

- sketch.pde: Fichero de ejemplo del sketch para Arduino del sistema a controlar.

El código de apoyo para el sistema remoto se encuentra en **practica_1B _vxworks_2020-v1.zip**. Este contiene los siguientes ficheros:

- displayB.c: Incluye el código principal del display del apartado B.
- displayB.h: Incluye la cabecera de las funciones de control de display del apartado B.
- controladorB.c: Fichero de ejemplo para realizar el controlador del apartado B. Este fichero debe ser modificado y entregado como parte de la práctica.

Además se incluirá el código de apoyo de un modulo para conectar el sistema a controlar y el sistema remoto mediante un cable serie (**modulo_serie-v1.zip**). Este contiene los siguientes ficheros:

- Serialdll.dll: librería dinámica para Windows que permite comunicar el Arduino y el PC mediante un puerto serie COM.
- serialcall.c: Código de un modulo kernel que permite exportar la funcionalidad de la librería anterior al kernel de VxWorks.
- serialcallLib.c: Codigo que permite enlazar una aplicación VxWorks con el modulo de kernel anterior (a incluir en el programa del sistema remoto).
- serialcallLib.h: Fichero de cabeceras del código de enlace anterior (a incluir en el programa del sistema remoto).

Apartado C)

Se desea ampliar el sistema del apartado B para incluir un sistema de paradas automáticas para cargar/descargar el cemento..

1 Características del entorno a controlar:

El sistema a controlar es el mismo del apartado B pero con la inclusión de depósitos de cemento donde descargar parte del cemento. La carretilla deberá detectar la proximidad del depósito y reducir su marcha hasta detenerse debajo del deposito, después deberá esperar hasta que se termine la descarga del cemento, en cuyo momento deberá reanudar su marcha.

Para controlar dicho sistema se incluyen los siguientes cambios:

- Se incorpora una funcionalidad extra al GPS de forma que, además de la velocidad, indicará la distancia que queda al próximo depósito.
- Se incorporará un sensor acoplado al depósito que indicará cuando el proceso de descarga ha terminado.

Los objetivos a conseguir serán los siguientes:

- El sistema deberá ejecutar normalmente (como en el apartado B) hasta que la distancia hasta el deposito se haya reducido hasta el mínimo aceptable. En ese momento deberá cambiarse el modo de ejecución para comenzar el proceso de frenado. Dicho proceso implica cambios respecto al modo de funcionamiento normal:
    - El proceso de ajuste de la velocidad deberá duplicar su frecuencia de uso para facilitar la precisión del frenado.
    - Los focos deberán mantenerse encendidos constantemente para indicar la detención de la carretilla. (No será necesario por tanto comprobar si hay penumbra o no).
    - El mezclador debe funcionar igual que antes.
- El proceso de frenado terminara cuando la distancia al deposito sea 0 y la velocidad de llegada sea menor de 10 m/s (En otro caso la descarga no se producirá). En ese momento el freno de emergencia del depósito parará la carretilla (sin importar el acelerador ni el freno) y cambiará el modo de ejecución para comenzar el proceso de descarga. En este modo los cambios son los siguientes.
    - Debe detectarse el momento de finalizar la descarga. Esto debe realizarse con la máxima frecuencia posible.
    - No es necesario comprobar la velocidad ni activar el freno ni el acelerador.
    - Los focos deberán mantenerse encendidos constantemente para indicar la detención de la carretilla. (No será necesario por tanto comprobar si hay penumbra o no).
    - El mezclador debe funcionar igual que antes.
- Tan pronto la descarga finalice, el acelerador y el freno volverán a funcionar y se volverá al modo de ejecución normal.

2 Definición del sistema a controlar:

Al igual que en el apartado B el sistema de control consta de dos partes diferenciadas:

- El sistema electrónico de control que esta instalado en la propia carretilla
- El sistema de control remoto instalado en un servidor autónomo del centro de control.

Ambos sistemas se comunican utilizando una conexión serie UART estándar.

2.1 Definición del sistema electrónico de control de la carretilla:

Los requisitos del sistema electrónico de la caretilla será los del apartado B pero un sistema de selección de la distancia al próximo deposito junto con un sistema activación de la distancia elegida y de fin de parada. En cualquier caso su funcionamiento dependerá del modo de operación que tenga el sistema electrónico en cada momento. Los modos de operación son los siguientes:

2.1.1 Modo de selección de distancia:

En este modo el sistema funcionara igual que el apartado B añadiendo las siguientes tareas y modificaciones a las tareas antiguas.

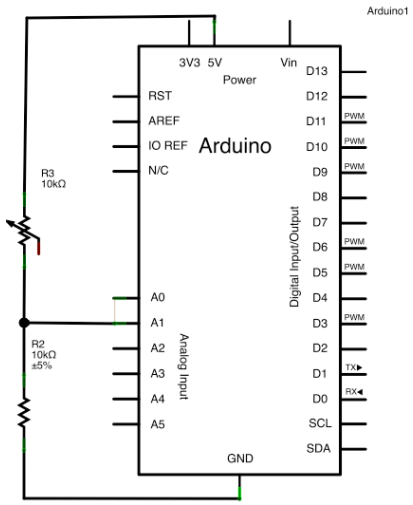
2.1.1.1 Modificación de servidor de comunicaciones

El servidor de comunicaciones deberá modificarse para aceptar una nueva petición que diga si la carretilla se esta moviendo o esta parada (en este modo siempre esta moviéndose).
La siguiente tabla incluye todos los mensajes del protocolo para las nuevas tareas.

Petición	Mensaje petición	Mensaje respuesta	Mensaje error
Leer EstadoMovimiento	STP:<SP>REQ<CR>	STP:<SP><SP>GO<CR>	MSG:<SP>ERR<CR>

2.1.1.2 Selector de la distancia al deposito:

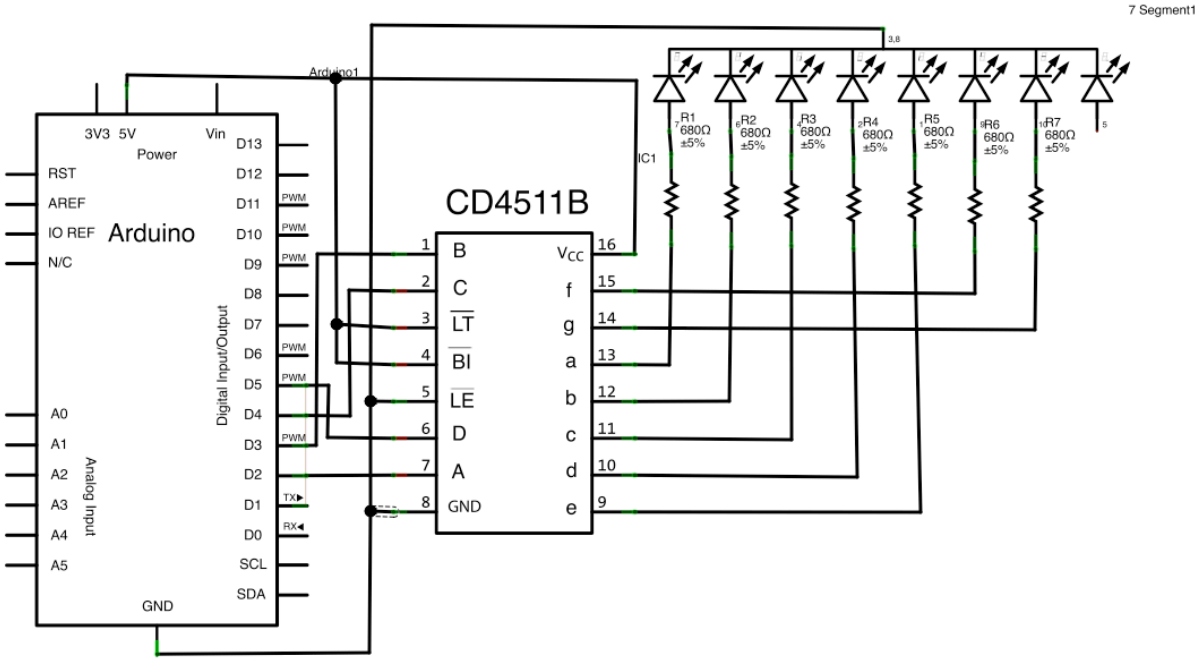
El Selector de la distancia al depósito se implementa mediante una potenciómetro para recibir el valor de la distancia. El esquema del circuito necesario es el siguiente.:



Deberá programarse una tarea periódica en el microcontrolador que almacene en cada periodo el valor actual que tiene el sensor. El valor deberá estar comprendido entre 10000 y 90000 metros. El alumno deberá elegir un valor apropiado para el periodo.

2.1.1.3 Display de la distancia seleccionada:

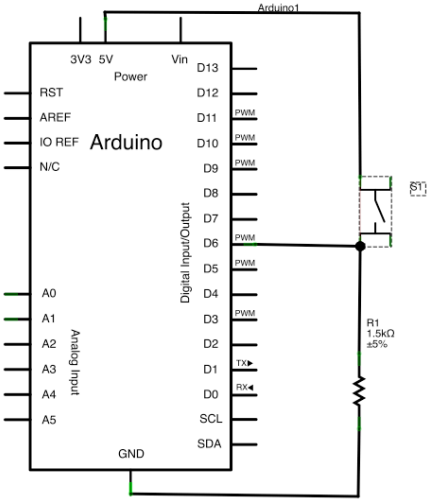
Se utiliza un display de siete segmentos de cátodo común junto con un chip 4511 para convertir de binario a las señales del 7 segmentos. El esquema del circuito es el siguiente.



Deberá programarse una tarea periódica en el microcontrolador que escriba en el display el valor mas reciente leído por el selector de distancia (en decenas de miles de metros).
El alumno deberá elegir un valor apropiado para el periodo.

2.1.1.4 Validación de la distancia al deposito:

El sistema de validación de la distancia al depósito se implementa mediante una botón para dar por bueno el valor actual de la distancia. El esquema del circuito necesario es el siguiente.:



Deberá programarse una tarea periódica en el microcontrolador que compruebe cuando se ha producido una pulsación del botón (apretar y soltar el botón). En ese momento en apuntara el valor actual del selector de distancia como la distancia actual y además se cambiara al siguiente modo. El alumno deberá elegir un valor apropiado para el periodo.

2.1.2 Modo de acercamiento al deposito:

En este modo el sistema funcionara igual que el apartado B añadiendo las siguientes tareas y modificaciones a las tareas antiguas.

2.1.2.1 Display de la distancia real al deposito:

El display de la distancia al depósito calculará (usando la velocidad) cual es la distancia actual al deposito. Si la distancia calculada es igual o menor que 0 y la velocidad actual es igual o menor de 10 m/s se asignara un cero a la distancia actual y se cambiara al modo de parada. Si la distancia es menor o igual a 0 y la velocidad es mayor que 10m/s se cambiara al modo anterior de selección de distancia. El hardware usado será el mismo de la tarea del display de la distancia seleccionada. Deberá programarse una tarea periódica en el microcontrolador para esta tarea. El alumno deberá elegir un valor apropiado para el periodo.

2.1.2.2 Modificación de servidor de comunicaciones

El servidor de comunicaciones deberá modificarse para aceptar una nueva petición que solicite el valor actual de la distancia, así como otra petición que diga si la carretilla se esta moviendo o esta parada (en este modo siempre esta moviéndose)
La siguiente tabla incluye todos los mensajes del protocolo para las nuevas tareas.

Petición	Mensaje petición	Mensaje respuesta	Mensaje error
Leer EstadoMovimiento	STP:<SP>REQ<CR>	STP:<SP><SP>GO<CR>	MSG:<SP>ERR<CR>
Leer Distancia	DS:<SP><SP>REQ<CR>	DS:00000<CR>	MSG:<SP>ERR<CR>

2.1.3 Modo de parada:

En este modo el sistema funcionara igual que el apartado B añadiendo las siguientes tareas y modificaciones a las tareas antiguas.

2.1.3.1 Modificación de la velocidad:

En este modo la velocidad debe ser siempre 0. Pase lo que pase. Deberá programarse una tarea periódica en el microcontrolador para esta tarea. El alumno deberá elegir un valor apropiado para el periodo.

2.1.3.2 Modificación de servidor de comunicaciones

El servidor de comunicaciones deberá incluirá una nueva petición que diga si la carretilla se esta moviendo o esta parada (en este modo siempre esta parada).
La siguiente tabla incluye todos los mensajes del protocolo para las nuevas tareas.

Petición	Mensaje petición	Mensaje respuesta	Mensaje error
Leer EstadoMovimiento	STP:<SP>REQ<CR>	STP:STOP<CR>	MSG:<SP>ERR<CR>

2.1.3.3 Lectura de fin de parada:

Esta tarea deberá usar el mismo hardware (el botón) utilizado para la tareas de validación de la parada. Cuando se produzca una pulsación del botón (apretar y soltar el botón) debe cambiarse al modo de selección de distancia. Deberá programarse una tarea periódica en el microcontrolador para esta tarea. El alumno deberá elegir un valor apropiado para el periodo.

2.2 Definición del sistema remoto:

El sistema remoto debe implementar los 3 modos indicados en la sección 1 (características del entorno a controlar). Estos modos de operación (normal, frenado y descarga) utilizarán las tareas que necesitan del apartado B (con los periodos que se requieran) y además podrán requerir las siguientes tareas:

2.2.1 Lectura de la distancia al deposito:

El sistema remoto deberá implementar una tarea periódica que envíe una petición a la carretilla para recibir el valor actual de la distancia al deposito y mostrará el valor por el display utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

2.2.2 Lectura del estado de movimiento de la carretilla:

El sistema remoto deberá implementar una tarea periódica que envíe una petición a la carretilla para recibir el valor actual del Estado de movimiento de la carretilla (parada, en movimiento) y mostrará el valor por el display utilizando la función correspondiente.
El alumno deberá calcular el periodo correcto para esta tarea.

Las funciones de acceso al display del servidor remoto son las siguientes:

ELEMENTO	***FUNCIÓN***	***TIEMPO DE COMPUTO***
Distancia	displayDistance (int distance)	0.5 Segundos
Fin Parada	displayStop (int Stop)	0.5 Segundos

Se pide:

5. Diseñar un planificador cíclico que permita controlar el sistema descrito anteriormente Para:
    e) El sistema remoto
    f) El control electrónico de la carretilla

6. Implementar dicho planificador cíclico para:
    e) El sistema remoto utilizando el SSOO VxWorks y el código de apoyo
    f) El control electrónico de la carretilla utilizando el microcontrolador Arduino UNO y hardware de apoyo.

El código de apoyo para el sistema a controlar se encuentra en **practica_1_sketch_2020-v1.zip**. Este contiene los siguientes ficheros:

- sketch.pde: Fichero de ejemplo del sketch para Arduino del sistema a controlar.

El código de apoyo para el sistema remoto se encuentra en **practica_1C_vxworks_2020-v1.zip**. Este contiene los siguientes ficheros:

- displayC.c: Incluye el código principal del display del apartado C.
- displayC.h: Incluye la cabecera de las funciones de control de display del apartado C.
- controladorC.c: Fichero de ejemplo para realizar el controlador del apartado C. Este fichero debe ser modificado y entregado como parte de la práctica.

Además se incluirá el código de apoyo de un modulo para conectar el sistema a controlar y el sistema remoto mediante un cable serie (**modulo_serie-v1.zip**). Este contiene los siguientes ficheros:

- Serialdll.dll: librería dinámica para Windows que permite comunicar el Arduino y el PC mediante un puerto serie COM.
- serialcall.c: Código de un modulo kernel que permite exportar la funcionalidad de la librería anterior al kernel de VxWorks.
- serialcallLib.c: Codigo que permite enlazar una aplicación VxWorks con el modulo de kernel anterior (a incluir en el programa del sistema remoto).
- serialcallLib.h: Fichero de cabeceras del código de enlace anterior (a incluir en el programa del sistema remoto).

Apartado D)

Se desea ampliar el sistema del apartado C para incluir un modo de emergencia en caso de un fallo en los tiempos del servidor remoto.

1 Características del entorno a controlar:

El sistema a controlar es el mismo del apartado C. En este sistema las funciones de acceso al display pueden sufrir retraso mas allá del tiempo de computo máximo previsto. En caso que se produzca un error en los tiempos de respuesta el sistemar remoto debe ejecutarse un modo de parada segura que consiste en lo siguiente.

- La carretilla debe detenerse. Para ello el freno debe ejecutarse constantemente y el acelerador debe estar desactivado.
- Los focos deberán mantenerse encendidos constantemente para indicar un estado de emergencia.
- El mezclador debe funcionar igual que antes.
- Avisa al sistema eléctrico de la carretilla que entre en un modo de parada de emergencia equivalente.

2 Definición del sistema a controlar:

Al igual que en el apartado C el sistema de control consta de dos partes diferenciadas:

- El sistema electrónico de control que esta instalado en la propia carretilla
- El sistema de control remoto instalado en un servidor autónomo del centro de control.

Ambos sistemas se comunican utilizando una conexión serie UART estándar.

2.1 Definición del sistema electrónico de control de la carretilla:

Los requisitos del sistema electrónico de la caretilla será los del apartado C pero incluirá un modo de parada de emergencia:

Además la tarea del servidor de comunicaciones deberá ser modificada en todos los modos para incluir la petición de salto al modo de emergencia.

2.1.1 Modo de parada de emergencia:

En este modo el sistema incluirá las mismas tareas del apartado B excepto la tarea de lectura del sensor de luminosidad. Además se modificará la tarea de los focos para que luzcan continuamente, la tarea del acelerador para que este desactivado todo el tiempo, y la tarea del freno para que este activado todo el tiempo.

2.1.2 Modificación de servidor de comunicaciones

El servidor de comunicaciones deberá incluirá una nueva petición para cambiar al modo de parada de emergencia.
La siguiente tabla incluye todos los mensajes del protocolo para las nuevas tareas.

Petición	Mensaje petición	Mensaje respuesta	Mensaje error
Activar modo emergencia	ERR:<SP>SET<CR>	ERR:<SP><SP>OK<CR>	MSG:<SP>ERR<CR>

2.2 Definición del sistema remoto:

El sistema remoto debe implementar el modo de emergencia indicado en la sección 1 (características del entorno a controlar).

Se pide:

7. Diseñar un planificador cíclico que permita controlar el sistema descrito anteriormente Para:
    g) El sistema remoto
    h) El control electrónico de la carretilla

8. Implementar dicho planificador cíclico para:
    g) El sistema remoto utilizando el SSOO VxWorks y el código de apoyo
    h) El control electrónico de la carretilla utilizando el microcontrolador Arduino UNO y hardware de apoyo.

El código de apoyo para el sistema a controlar se encuentra en **practica_1_sketch_2020-v1.zip**. Este contiene los siguientes ficheros:

- sketch.pde: Fichero de ejemplo del sketch para Arduino del sistema a controlar.

El código de apoyo para el sistema remoto se encuentra en **practica_1D_vxworks_2020-v1.zip**. Este contiene los siguientes ficheros:

- displayD.c: Incluye el código principal del display del apartado D.
- displayD.h: Incluye la cabecera de las funciones de control de display del apartado D.
- controladorD.c: Fichero de ejemplo para realizar el controlador del apartado D. Este fichero debe ser modificado y entregado como parte de la práctica.

Además se incluirá el código de apoyo de un modulo para conectar el sistema a controlar y el sistema remoto mediante un cable serie (**modulo_serie-v1.zip**). Este contiene los siguientes ficheros:

- Serialdll.dll: librería dinámica para Windows que permite comunicar el Arduino y el PC mediante un puerto serie COM.
- serialcall.c: Código de un modulo kernel que permite exportar la funcionalidad de la librería anterior al kernel de VxWorks.
- serialcallLib.c: Codigo que permite enlazar una aplicación VxWorks con el modulo de kernel anterior (a incluir en el programa del sistema remoto).
- serialcallLib.h: Fichero de cabeceras del código de enlace anterior (a incluir en el programa del sistema remoto).

- ## Documentación a entregar

Para entregar la práctica deberá ejecutarse el entregador Web situado en la pagina de la signatura En aula global 2 en el apartado entrega de practicas.

En caso de que se entregue la práctica varias veces sólo será válida la última entrega.

Los ficheros a entregar son los siguientes:

memoria.txt | memoria.pdf | memoria.doc
Memoria de la práctica.
controladorA.c
Código fuente del controlador del apartado A.
sketchA.ino | sketchA.pde
Código fuente del sketch de arduino del apartado A.
controladorB.c
Código fuente del controlador del apartado B.
sketchB.ino | sketchB.pde
Código fuente del sketch de arduino del apartado B.
controladorCc
Código fuente del controlador del apartado C.
sketchC.ino | sketchC.pde
Código fuente del sketch de arduino del apartado C.
controladorD.c
Código fuente del controlador del apartado D.
sketchD.ino | sketchD.pde
Código fuente del sketch de arduino del apartado D.

Además de la entrega electrónica será necesaria una corrección presencial de la practica para ver su correcto funcionamiento. Dicha corrección se podrá realizar en cualquier clase de practicas o en horario de tutorías, siempre antes de la fecha de entrega.

NOTA: Para la corrección presencial hay que presentar la configuración hardware completa y luego cargar los códigos de cada uno de los apartados. No es necesario presentar la memoria para la corrección presencial.

Plazo de entrega

Viernes 01 de Noviembre de 2019.